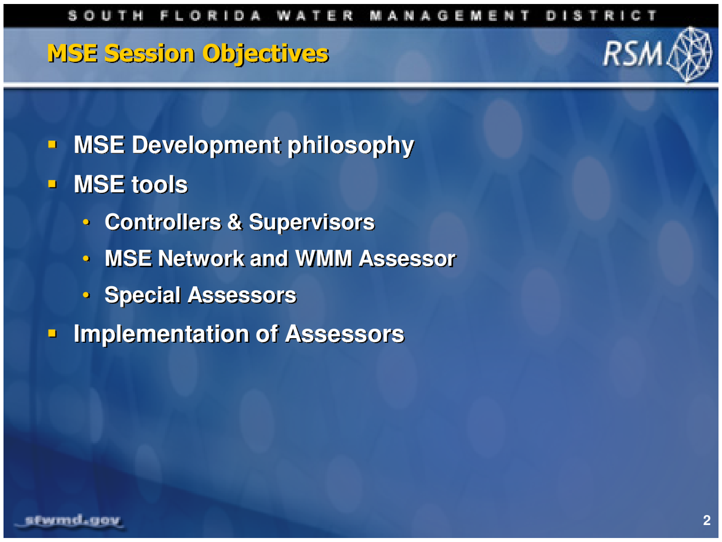
Lecture 15: Management Simulation Engine (MSE)

NOTE:

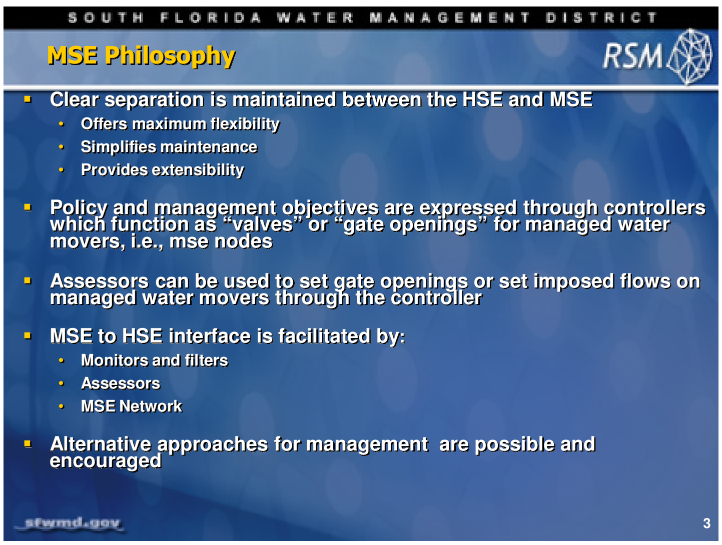
Materials required for this session

Additional Resources

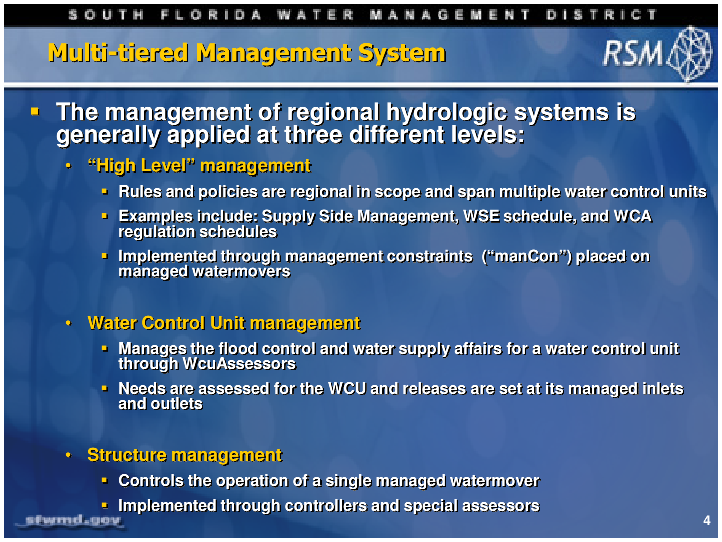
Related files



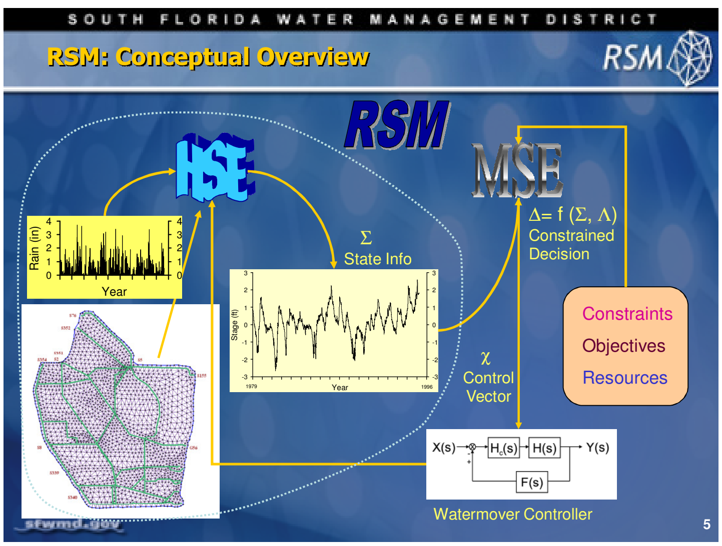
The session will be a discussion of these topics.



It was decided early in the development that the HSE and the MSE should be constructed separately. That would provide the greatest flexibility and simplify the maintenance. The HSE is a comprehensive hydrologic model by itself. The water management policies and management objectives are imposed on the HSE by setting the constraints implemented through controllers. There are a collection of assessors that determine how the structures can determine what the flow ought to be. There is a good connection between the HSE and the MSE through monitors and assessors. The MSE provides for a high degree of flexibility in implementing alternative approaches for management.



Management of the regional CSFFC project is a complex problem. It can be broken down into management at three different levels. At the highest level, the rules and policies are implemented at a high level broad-based regional approach through the application of constraints on how much water can pass through various structures. At the second level, flood control and water supply is managed at the water control unit (WCU) level. This is done by using assessors to determine the needs of the WCU and setting the flows at the inlets and outlets of the WCU. At the narrowest scope, controllers or special assessors are used to control specific structures. The future work will consist of determining what needs to be managed, at what level it should be implemented and creating the appropriate MSE functions.



Conceptual overview of RSM.

Two primary components: HSE and MSE.

An early decision was made to completely separate the hydrologic computations from the management computations.

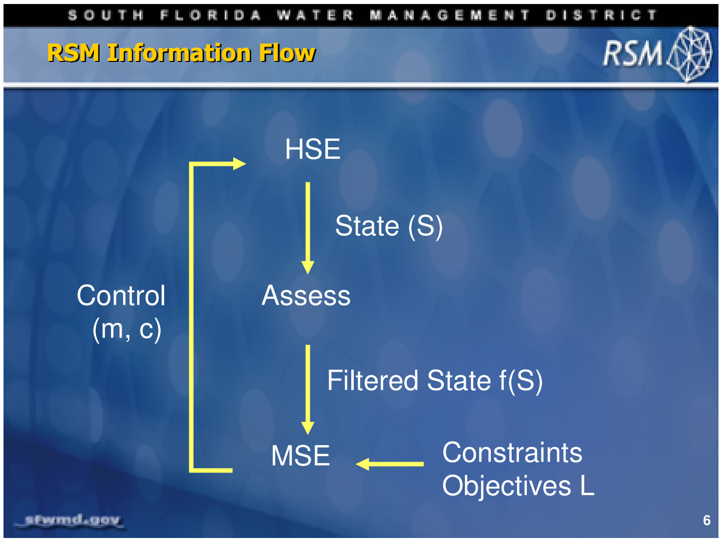
This allows the user to make many flexible changes to the system management without affecting the hydrologic computations.

HSE reads in boundary condition information, executes hydrologic computations which estimate state information.

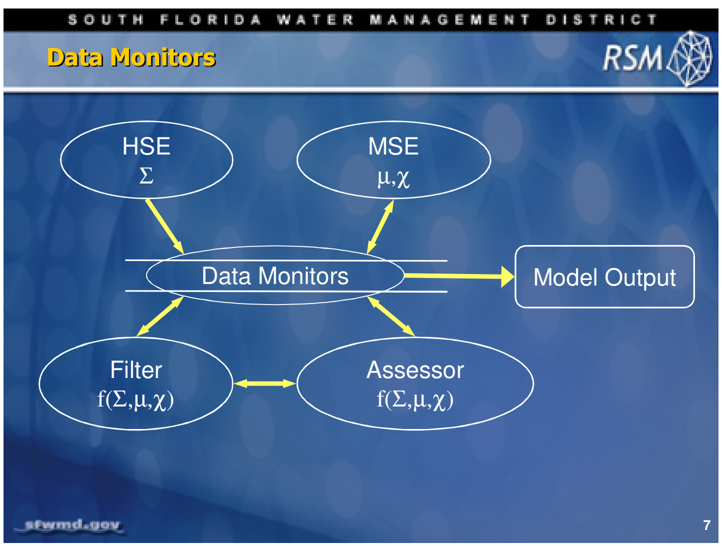
HSE controllers can react to state information, and enact watermover flow regulation within HSE.

MSE reads in constraint and objective boundary condition information, and state information from HSE.

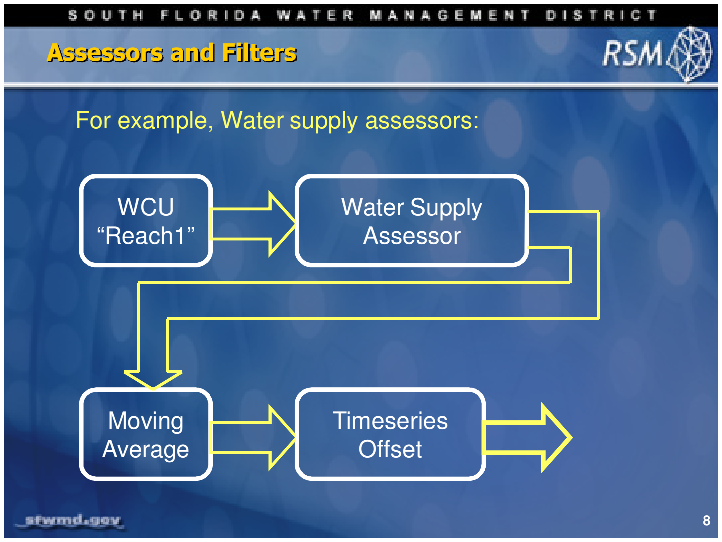
MSE produces a decision and control vector for HSE controllers.



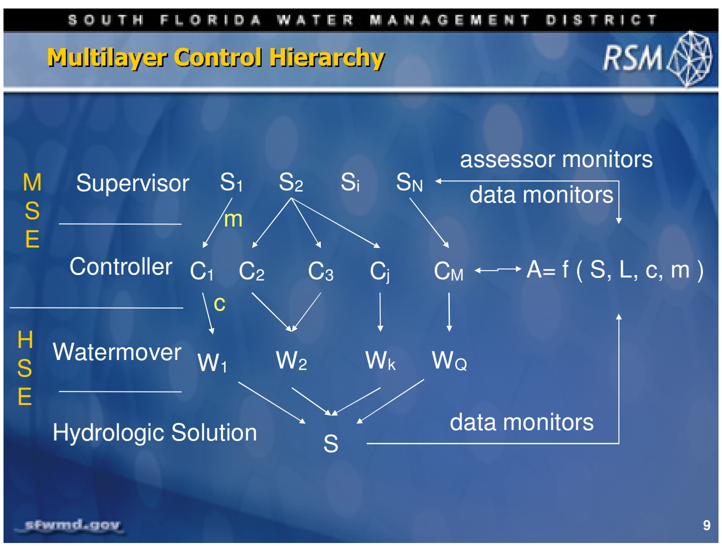
This is the information flow for the regional simulation model. The HSE provides the state information for the variables of interest including rainfall groundwater stage and canal stage. This information is passed to the MSE. The information may be the raw data or the data may be filtered by statistical analysis such as a moving average. The MSE takes the information from the state variables then applies certain rules, management constraints and controls the structures and water movers in the HSE. The management constraints include the regional water management policies as well as local structure management rules.



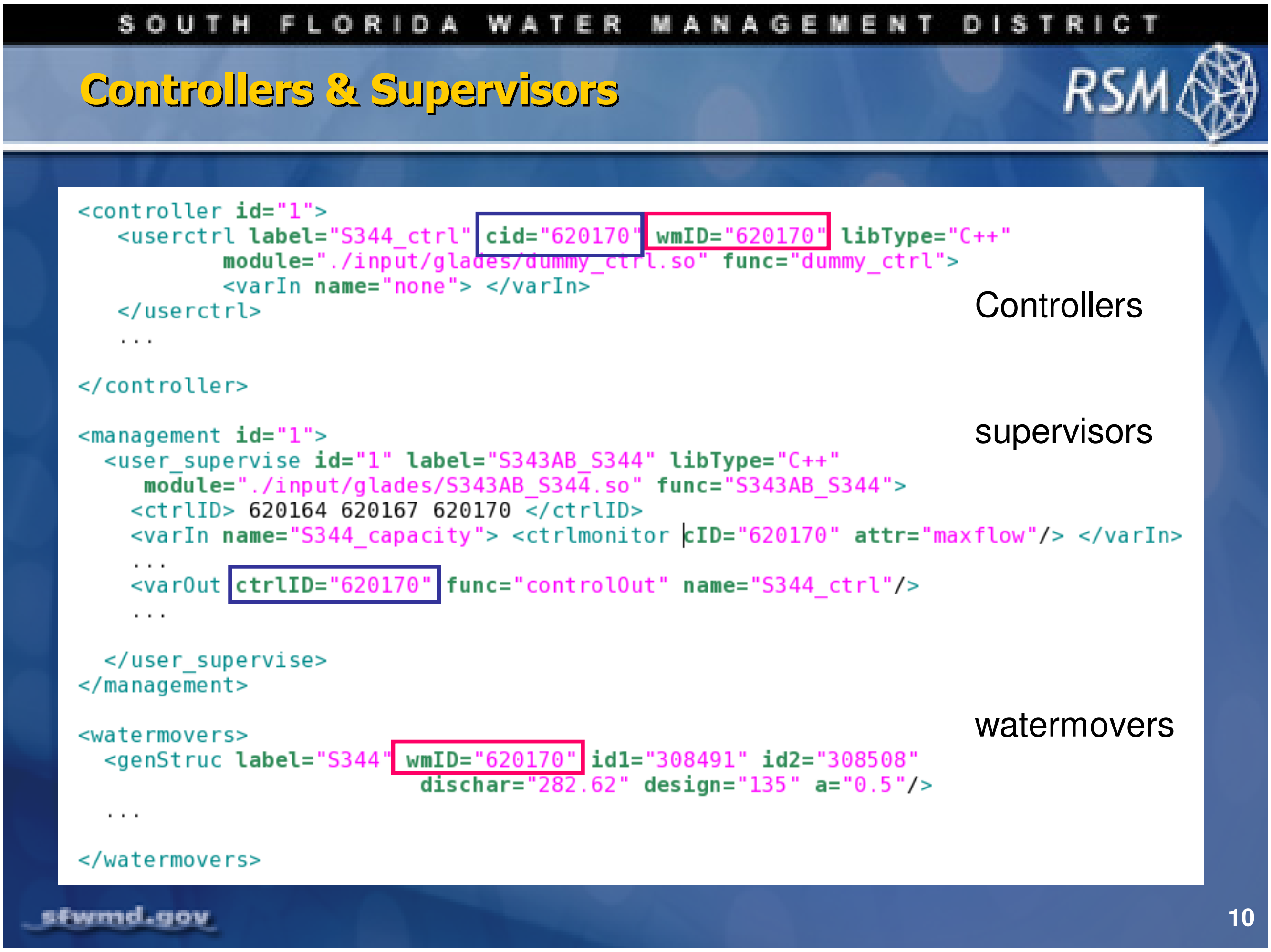
The data monitors are essential components of the interface between the HSE and the MSE. The monitors track the state information from the waterbodies, watermovers and MSE components as well as the output of filters and the assessors. The monitors produce the state information from waterbodies to the MSE as well as to the output. The MSE was created so that it uses the same data monitors as the HSE uses so no additional functionality had to be created for the MSE. The locations of the monitors are determined as part of the XML input.



The assessors are used to determine the volume of water in the selected WCUs for either food control or water supply. Generally, assessors can be used to assess any state or dynamic variable in the model. A filter is an extension of a monitor that allows you to perform operations on the values from assessors. The common filters include moving averages, average of a group of monitors, and monitor comparisons. The typical output is a time series of values to be used by the MSE.



The MSE is a multilayer system. At the lowest level we have the HSE with the watermovers that control the water flow between the waterbodies. There are various data monitors that tell us the state information at various times. There are various assessors that provide information about the state of the system such as controller or gate operations in addition to hydrologic data. The MSE is a multilayer at the system that allows supervisors control of multiple controllers and controllers to control multiple watermovers. The actions (A) taken by the MSE are a function of the state of the system (S), management constraints (L), the control settings (c), and management settings (m).



The controllers and supervisors are used to set the flow at specific structures based on local variables, seasonality or stage regulation schedules. Although controllers and supervisors cannot be used comprehensively due to computational expense, they are useful for selected structures.

The control functions are implemented using the user-defined controllers **<userctrl>**. These controllers are used where the control function is not a simple implementation of flow calculated elsewhere. The flow or control value is calculated in a C++ program and compiled into a library of control functions e.g. dummy.ctrl.so. The controllers are managed by a user_supervisor that controls one or more controllers **<varOut>** based on one or more inputs **<varIn>**. The controller in turn, controls a specific watermover. Each of these features occurs in separate XML blocks in the main run XML file.

SOUTH FLORIDA WATER MANAGEMENT DISTRICT

Controllers: User-defined

RSM

```
<userctrl label="S343A_ctrl" cid="620164" wmID="620164" libType="C++"
  module="./input/glades/dummy_ctrl.so" func="dummy_ctrl"> <varIn name="none"> </varIn>
</userctrl>
<userctrl label="S343B_ctrl" cid="620167" wmID="620167" libType="C++"
  module="./input/glades/dummy_ctrl.so" func="dummy_ctrl"> <varIn name="none"> </varIn>
</userctrl>

<userctrl label="S344_ctrl" cid="620170" wmID="620170" libType="C++"
  module="./input/glades/dummy_ctrl.so" func="dummy_ctrl"> <varIn name="none"> </varIn>
</userctrl>

<!-- ================================== SR29-1 ================================== -->
<userctrl label="SR29-1_ctrl" cid="660003" wmID="660003" libType="C++"
  module="./input/glades/SR29_seasonalctrl.so" func="SR29_seasonalctrl">
  <varIn name="year"    > <tkprmonitor                 attr="year" /> </varIn>
  <varIn name="month"   > <tkprmonitor                 attr="month"/> </varIn>
  <varIn name="day"     > <tkprmonitor                 attr="day"  /> </varIn>
  <varIn name="SR29_HW" > <segmentmonitor id="314270" attr="head" /> </varIn>
  <varIn name="str_name"  source="xml"> <string>SR29-1</string> </varIn>
</userctrl>

<userctrl label="SR29-4_ctrl" cid="660006" wmID="660006" libType="C++"
  module="./input/glades/SR29_seasonalctrl.so" func="SR29_seasonalctrl">
  <varIn name="year"    > <tkprmonitor                 attr="year" /> </varIn>
  <varIn name="month"   > <tkprmonitor                 attr="month"/> </varIn>
  <varIn name="day"     > <tkprmonitor                 attr="day"  /> </varIn>
  <varIn name="SR29_HW" > <segmentmonitor id="308295" attr="head" /> </varIn>
  <varIn name="str_name"  source="xml"> <string>SR29-4</string> </varIn>
</userctrl>
```

11

User defined controllers are specified in the controller block of the main run XML file. The specification can be a simple call to the controller C++ process program as with the controller for S343-344 structures or a more complex function that includes the necessary variables for the process program i.e., SR29 structures. The dummy_ctrl function simply passes the output from the supervisor to the controller.

SOUTH FLORIDA WATER MANAGEMENT DISTRICT

Used-Defined Controller

RSM

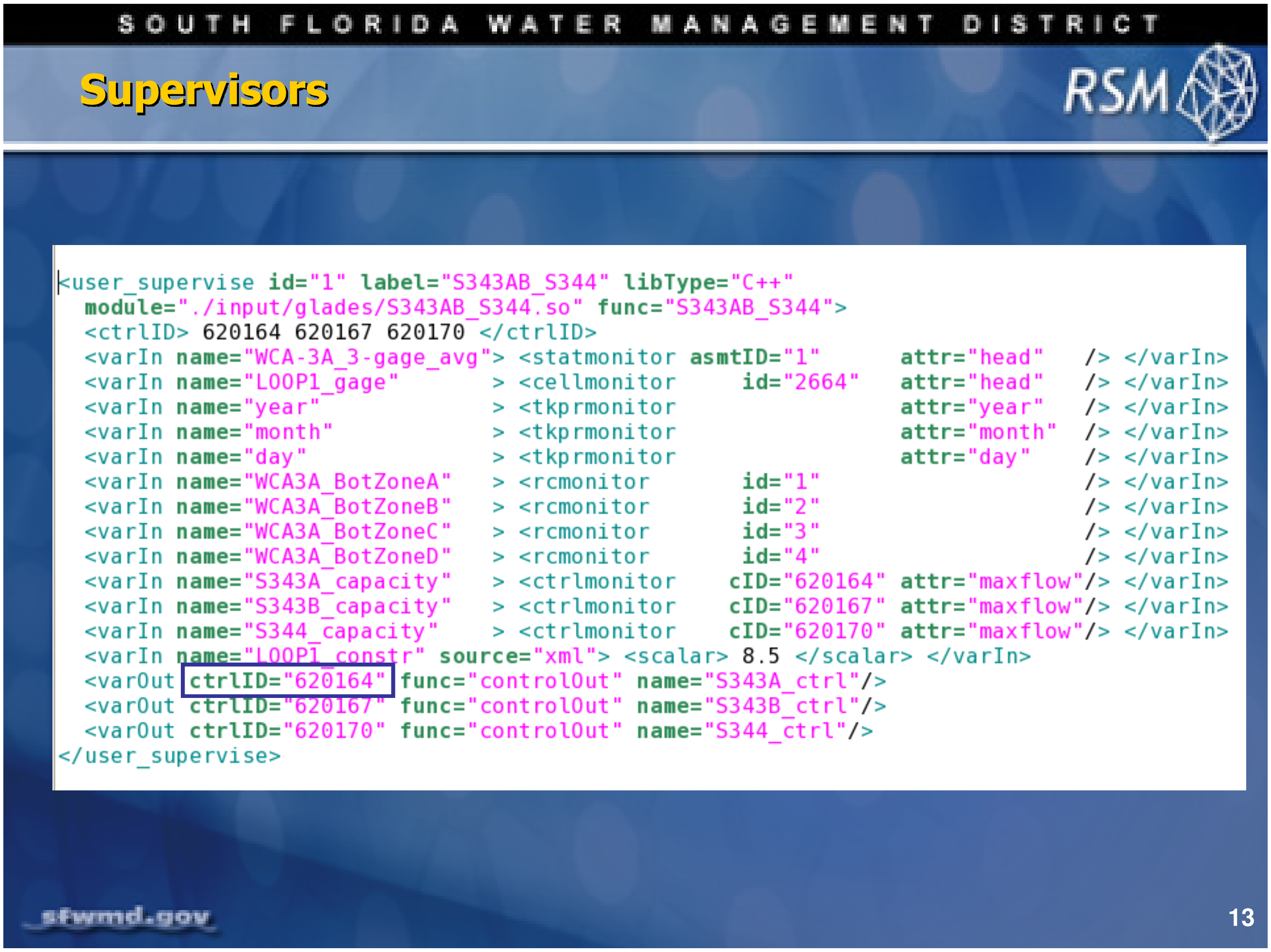
SR29_seasonalctrl.cc

```
try {

  // INPUT VARIABLE DECLARATION
  int    year      = (int) GetVarIn( func, "year",      lpInputStateMap );
  int    month     = (int) GetVarIn( func, "month",     lpInputStateMap );
  int    day       = (int) GetVarIn( func, "day",       lpInputStateMap );
  double SR29_HW   =       GetVarIn( func, "SR29_HW",   lpInputStateMap );
  string str_name  = XMLStringValue( func, "str_name",  lpInputStateMap );

  // INTERMEDIATE VARIABLE DECLARATION
  string structure_name = "none";

  // OUTPUT VARIABLE DECLARATION
  double controlOut = 0.;

  // simple sr29 operations that attempt to mimic what happens in reality that is:
  // open the main weirs in the wet season and close them during the dry season
  // rarely but occasionally they can be overtopped during dry season
  if ( month <= 5 or month >= 11 ) {
    controlOut = 0.0;
  }
  else {
    controlOut = 1.0;
  }

  return controlOut;
}
```

sfwmd.gov 12

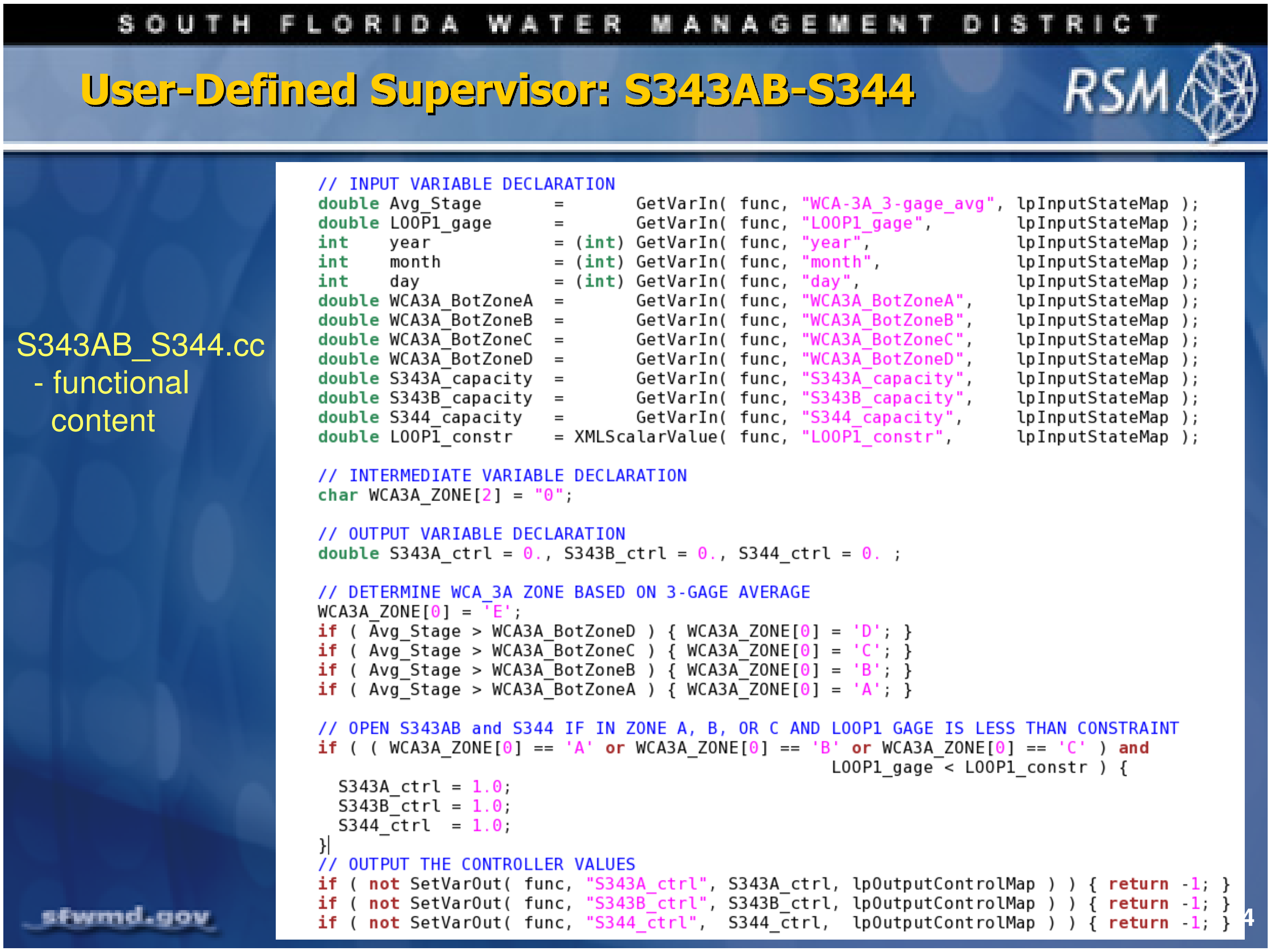
The user-controller process program for the S29 structures is simple, it only requires the data inputs from the model and sets the structure outflow to full flow during the wet season and closes the structure during the dry season. The source code is available in the **RSM/data/glades_lecsa/input/glades/SR29_seasonalctrl.cc** file. The structure control can be adjusted by changing the function in this C++ source code file and recompiling the function into a shared library using the following command, all on one line:

gcc SR29_seasonalctrl.cc -BSymbolic -fPIC -shared -o SR29_seasonalctrl.so -I../../../../trunk/src/ -I/opt/local/share2_64/include



The user-supervisor uses the information processed by the controller function to set the "**controlOut**" variable for each of the structures. The inputs for the supervisor include several monitors including the control monitors processed the user-controller, rule-curve monitors and the statmonitor that uses an assessor to average the water levels in three cells to obtain the average water level in WCA-3A.

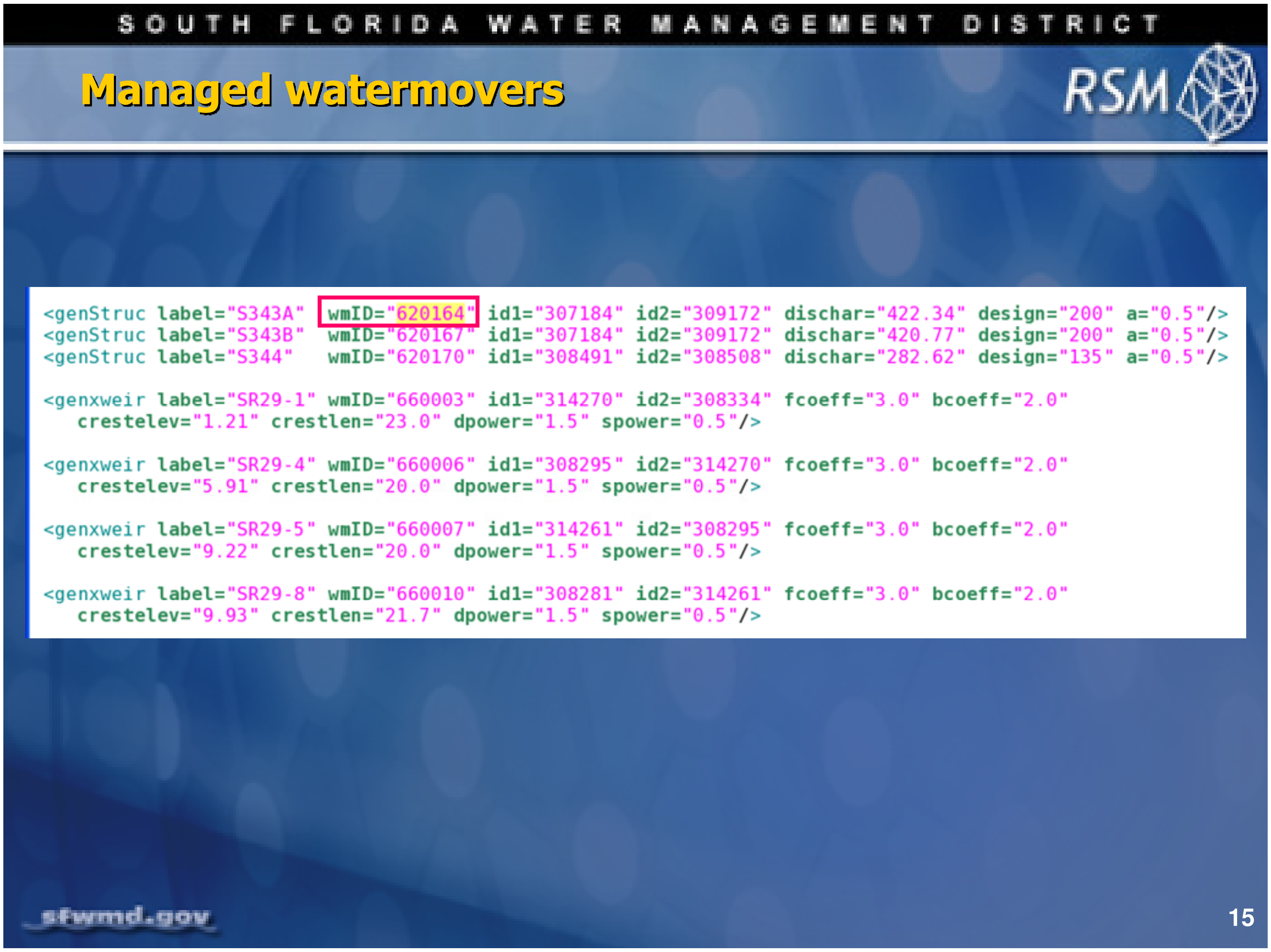
The high-lighted variable is the ID of the watermover that is controlled.



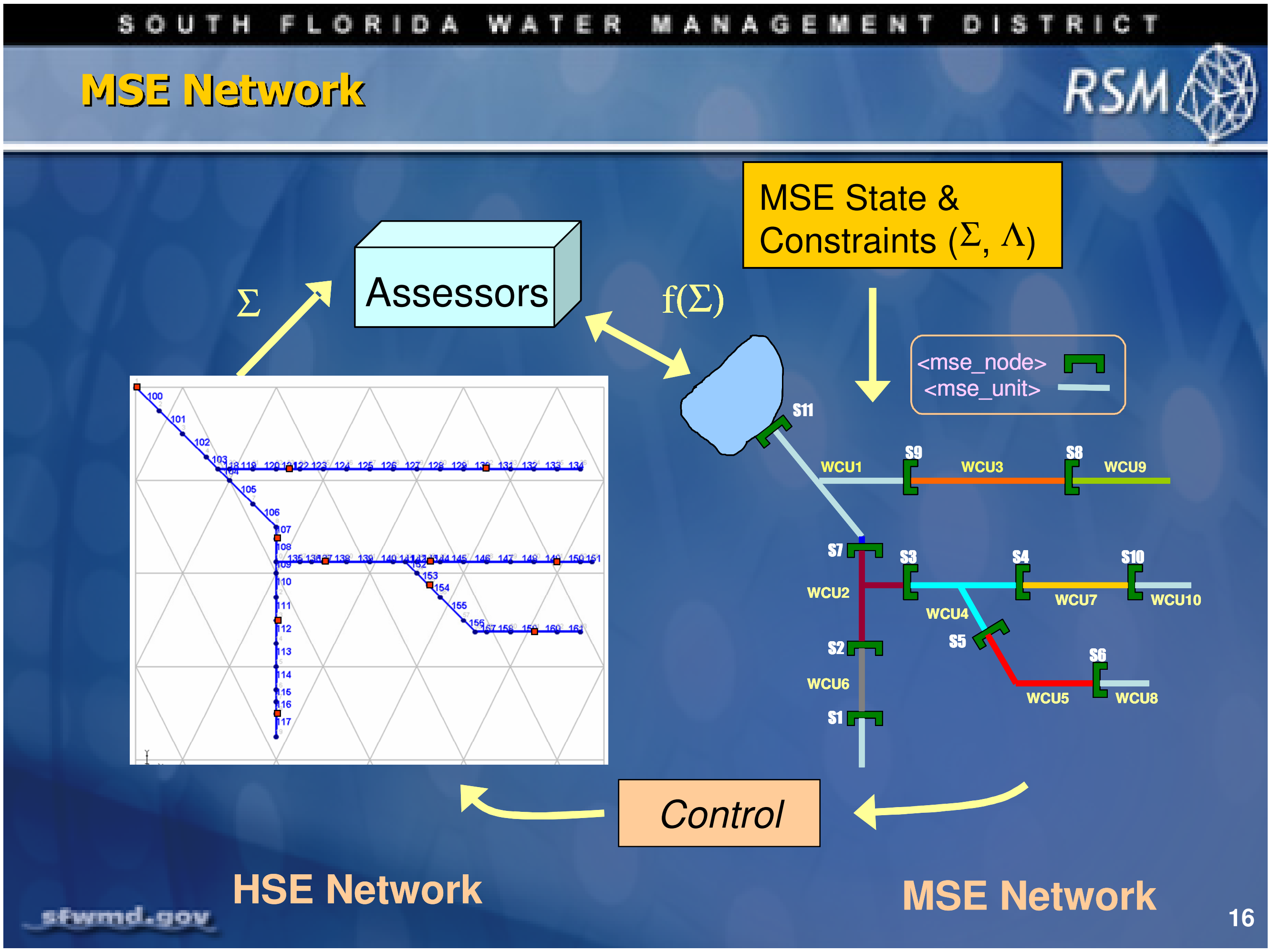
The user-supervisor for the S343-344 structures is more complicated than the one used for the S29 structures; it uses the WCA3A regulation stage schedule as well as the structure capacity to set the flow for the structures. Changes can be made to this controller source code file and recompiled using the following compile command:

g++ S343AB_S344.cc -BSymbolic -fPIC -shared -o S343AB_S344.so

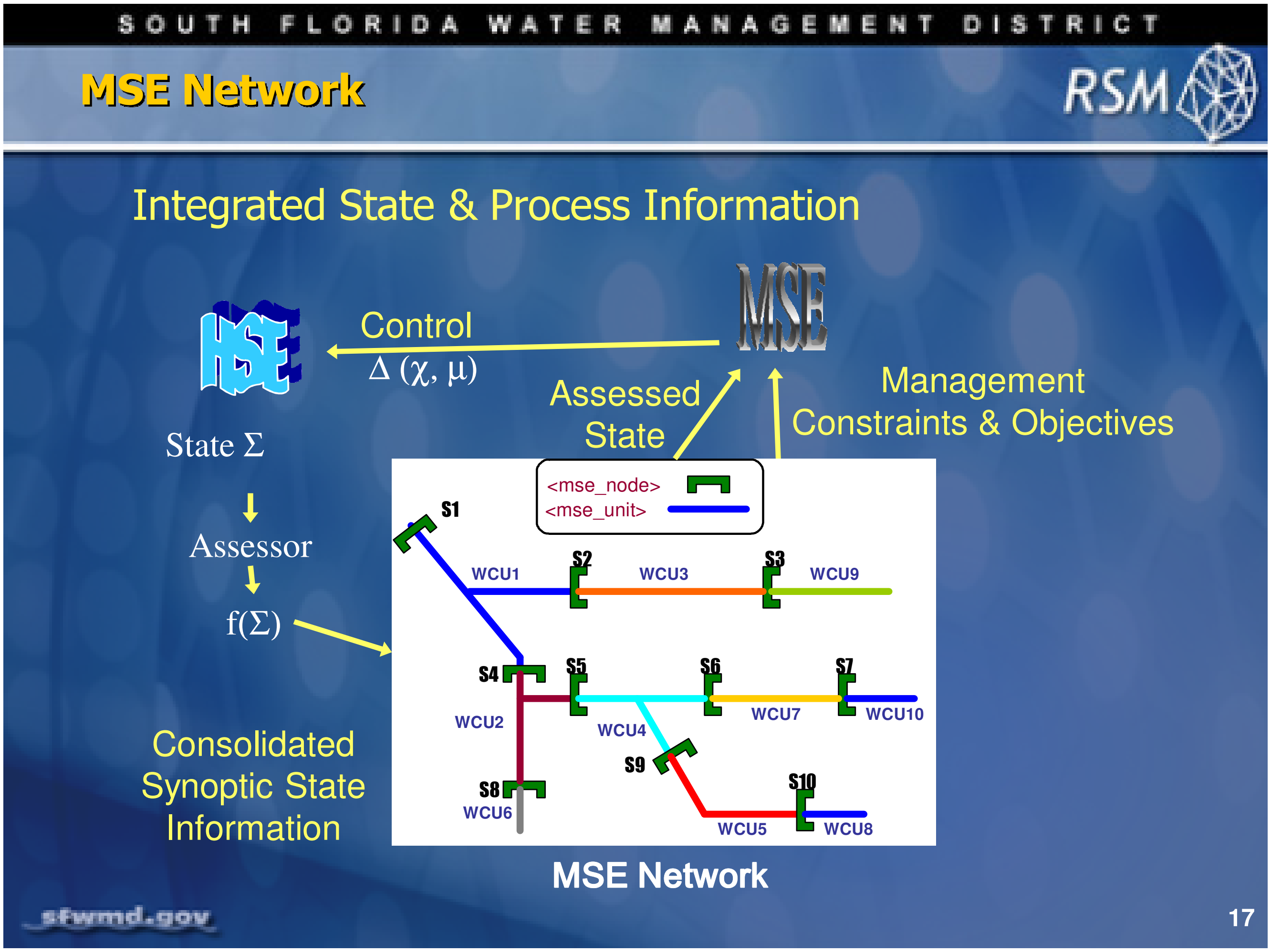
-I ../../../../trunk/src/ -I/opt/local/share2_64/include



The managed structures have watermovers where the flow is set by the controllers or the supervisors. The physical characteristics of the structure including the headwater and tailwater stages determine the flow capacity of the structure. The controller sets the flow within that capacity.

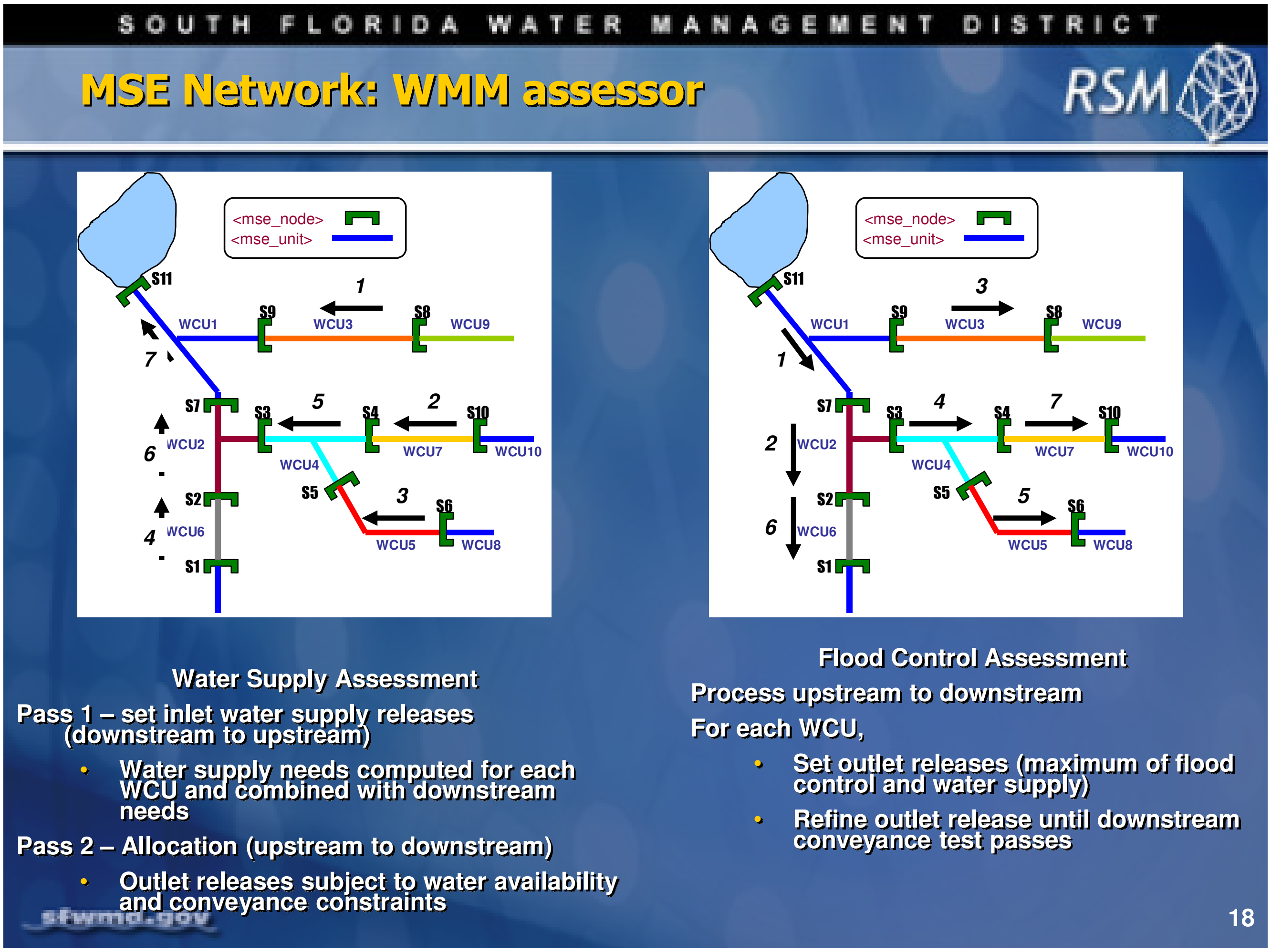


The MSE network is an abstraction of the HSE network to encapsulate the key components of the water control system without including the details of the HSE hydrology. The numbers of segments within the WCUs are unnecessary for managing the system. Provides a database on how you want the water management system to behave. This includes the flood control levels or water supply levels. Thus this information does not have to be carried by the HSE.

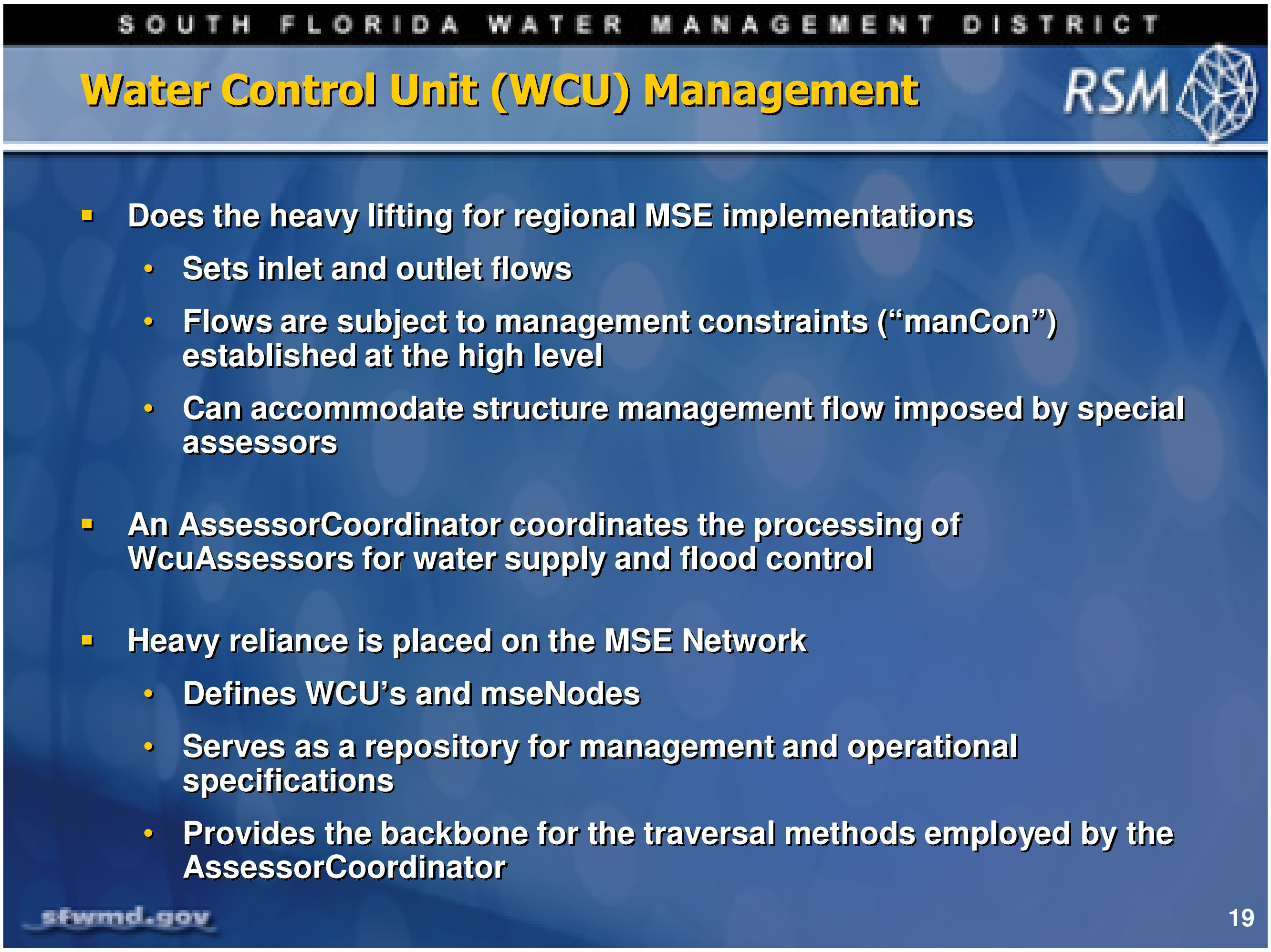


The MSE_network is an essential component of the MSE that links the components of the MSE together as well as linking the MSE to the HSE. The MSE network is an abstraction of the MSE information of the model and the hydrologic information in the model. To create an RSM-HSE you typically specify the canal network (segments) and mesh (cells) of the model. For the MSE to function it is not necessary to know the segments and cells. The MSE Network is an abstraction of the structures and the WCUs between the structures. The WCUs are composed of the segments that exist between the structures. The assessors take the state information from the HSE and provide consolidated state information for each WCU. This information, along with management objectives and constraints in the form of rules, is used to provide control information to the HSE.

The MSE network is specified in the xml. It is composed on nodes and WCU. The nodes are tied to structures in the HSE. The nodes have various properties that can be specified from the HSE and how you want them to behave. The WCUs are the various water bodies and control points and management constraints that describe how you want the system to behave.



This is an illustration of how the assessors are used to manage water supply (WS) and flood control (FC). This example, from Benchmark BM63, is a collection of fully defined WCUs that are canal reaches. The MSE_nodes are structures. This contains the details of the WCUs, the MSEnodes, the FC assessors and the WS assessors. The assessorCoordinator processes the FC assessors in the order in which they are presented in the xml file; order matters. They are processed in the order shown in the graphic on the right, from the upstream to downstream. After the FC assessors are applied, the WS assessors are processed from the downstream to the upstream inlet, determining the needs are to maintain water levels. This assessment determines the “needs”. A second pass is made for the WS assessors from the upstream to the downstream allocating the available water by various means based on priorities defined by the users. Finally, another pass is made from downstream to upstream assigning the appropriate flood control needs for each structure. The final assigned flow is the larger of the flood control or water supply flows. The way these assessments are applied to the HSE is through controllers. The controller assesses the nodemonitor and than the controller sets the flow at the structure appropriately. The details are available in the documentation.



The WCU is where the MSE primarily determines the flow subject to the known management constraints. It should be remembered that the WCU is a water body; a lake, collection of cells etc… The inlets and outlets of the WCUs are nodes that can be controlled by special assessors. There is an AssessorCoordinator that controls how the WCU behaves. The MSE Network is the backbone for the implementation of the WCUs and the AssessorCoordinator. The MSE Network is the repository for the rules, policies and constraints as well as the structure characteristics.

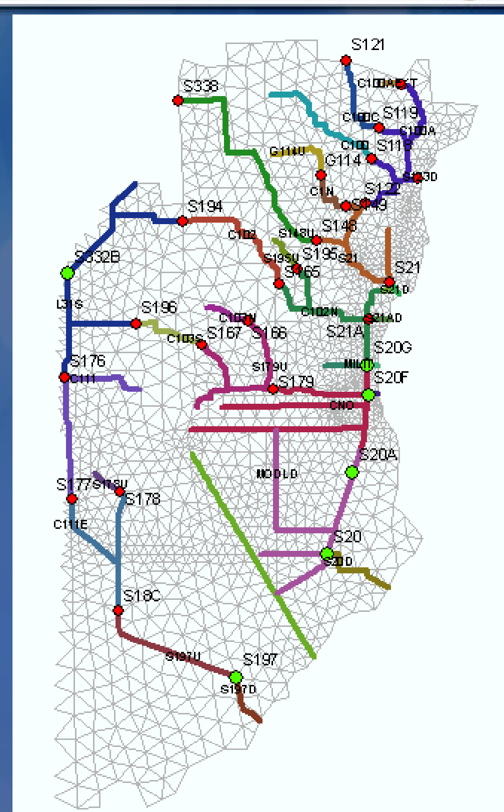
SOUTH FLORIDA WATER MANAGEMENT DISTRICT

Water Control Unit (WCU)

Typically, a collection of canal segments confined by MSE nodes that share the same operational constraints/ objectives.

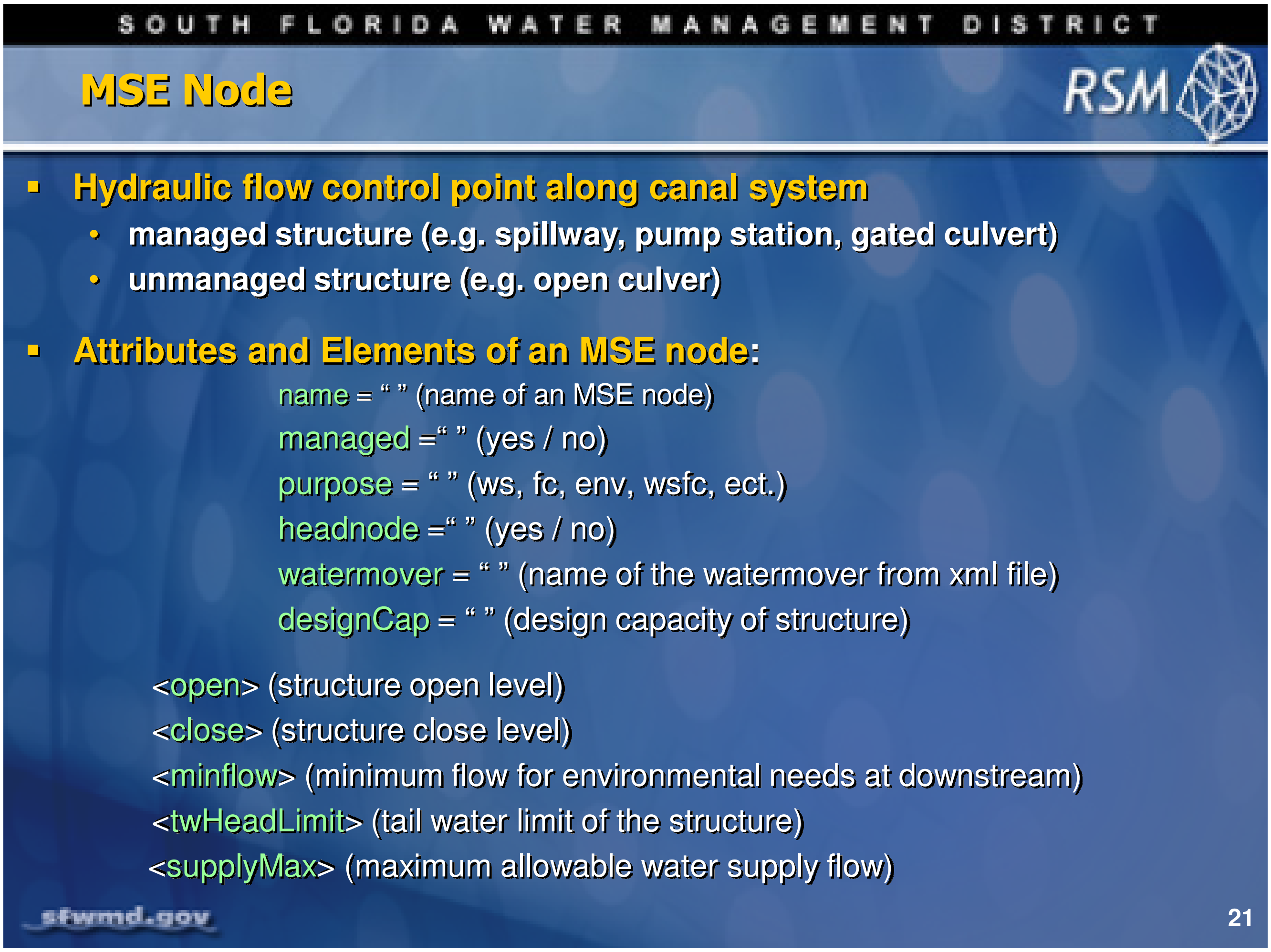
Elements of a WCU:

<hse_arcs>	list of segment ids
<hse_wbs>	list of waterbody ids
<wcds>	list of WCD ids
<lakes>	list of lakes ids
<basins>	list of basins ids
<impoundments>	list of impoundments ids
<inlet>	list of managed <mse_node>
<outlet>	list of managed <mse_node>
<maintLevel>	maintenance level for Water supply need
<fcLevel>	flood control level
<resLevel>	reserve level
<localLevel>	local level

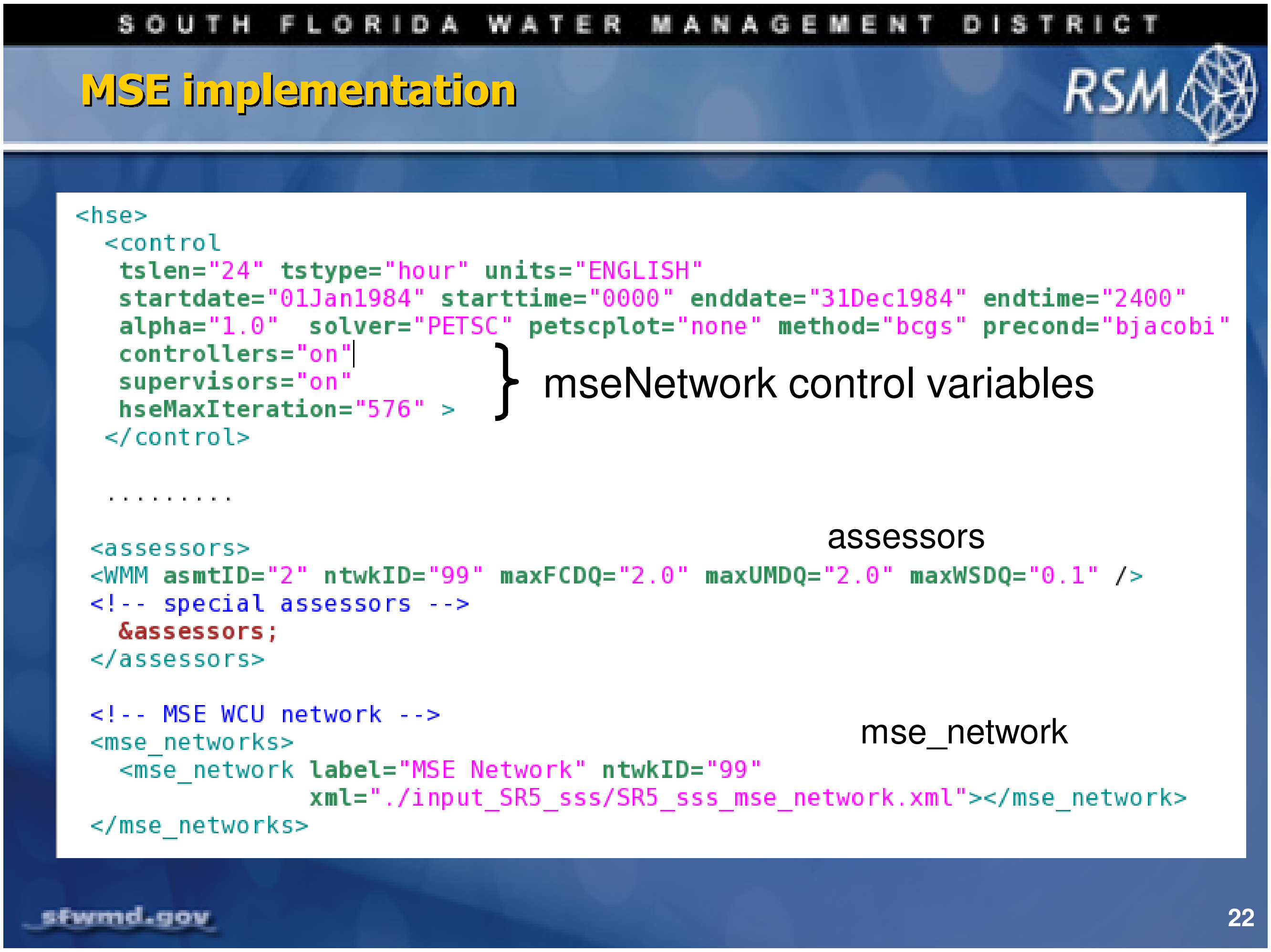


sfwmd.gov

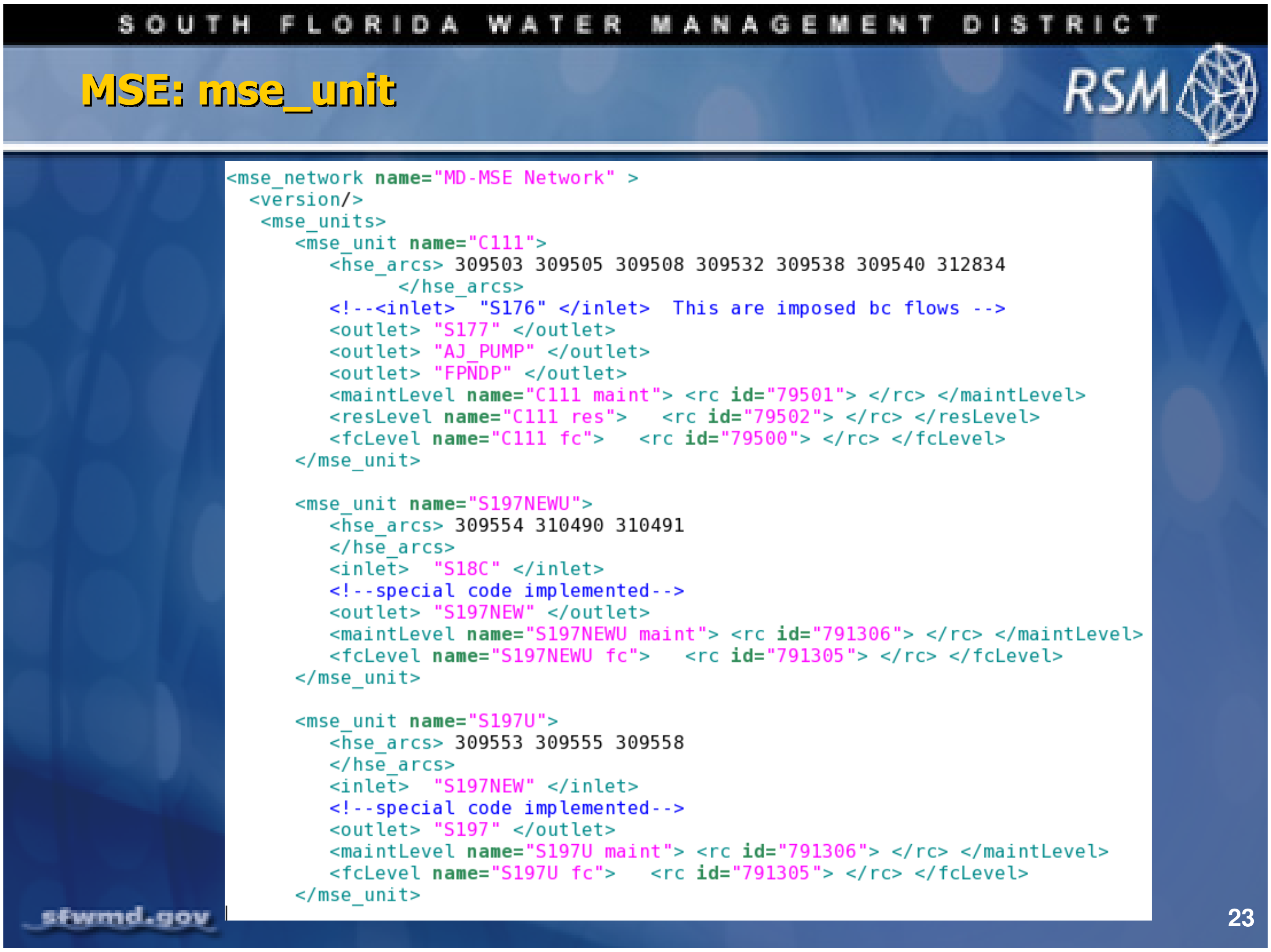
As an example, the Biscayne Bay coastal wetlands RSM is divided up into a number of WCUs. In this example, the WCUs are a collection of canal segments. Although the mesh cells drain to the WCUs, they are not defined as part of the WCUs. The WCUs can also be basins, cells, impoundments, WCDs or lakes. The elements include the canal segments and the inlet and outlet nodes. For each WCU it is necessary to define the flood control level, above which water is pumped out; maintenance level, the level that the WS assessors wish to maintain; the reserve level, below which water cannot be pumped out; and the local level, below which water cannot seep out or be used for ET.



The second part of the MSE network is composed of the MSE nodes. The MSE nodes are the control points between the WCUs. An MSE node is typically a managed structure where the flow is controlled by setting gate openings, turning on pumps or an unmanaged structure such as a culvert of a fixed head weir. The node can be managed for flood control, water supply environmental or other purposes. Defining the purpose implements default assessors and controllers for operating the structure (watermover). The elements of the **<mse_node>** determine specific conditions when the structure can be open or closed and flow restrictions. The hydrologic data and operating rules are obtained from the SFWMD Structure Book.



The implementation of the MSE requires several XML components. In the control block of the main XML file there are several attributes to control the MSE. The maximum iterations determine how long the MSE and the HSE will iterate until a solution is found. Remember, the MSE assessors are calculating estimated of the flood control and water supply requirements of the WCUs. The actual values will depend on the HSE solution for each structure for the Dheads. The required flow determined by the MSE assessors is likely to be different from the flow calculated at the structure and as a result the RSM will iterate between the MSE and HSE solutions. The main XML file will contain the mse_network element that specified the name of the network and the location of the mse_network definition XML file. The main XML file will also contain the assessors or the file containing the assessors. Typically, the MSE will use the default assessors for water supply and flood control; only special assessors are specified in the XML file. For the water supply and flood control assessors the maximum difference for water supply flows, flood control flows and unmanaged flows between the HSE and the MSE solution are specified.



The typical MSE network will contain several WCU defined by the mse_units. Each mse_unit will have a list of waterbodies, typically a list of the canal segments from the HSE network that are to be included in the mse_unit. The inlet and outlet nodes are identified along with the stage targets within the WCU. The minimum stage targets include the flood control level **<fcLevel>** above which drainage is required and the water level maintenance level **<maintLevel>** below which water is requested from upstream. In these three WCUs, the target levels are set using rule curves that implement different water level targets form the dry season and wetseason.

SOUTH FLORIDA WATER MANAGEMENT DISTRICT

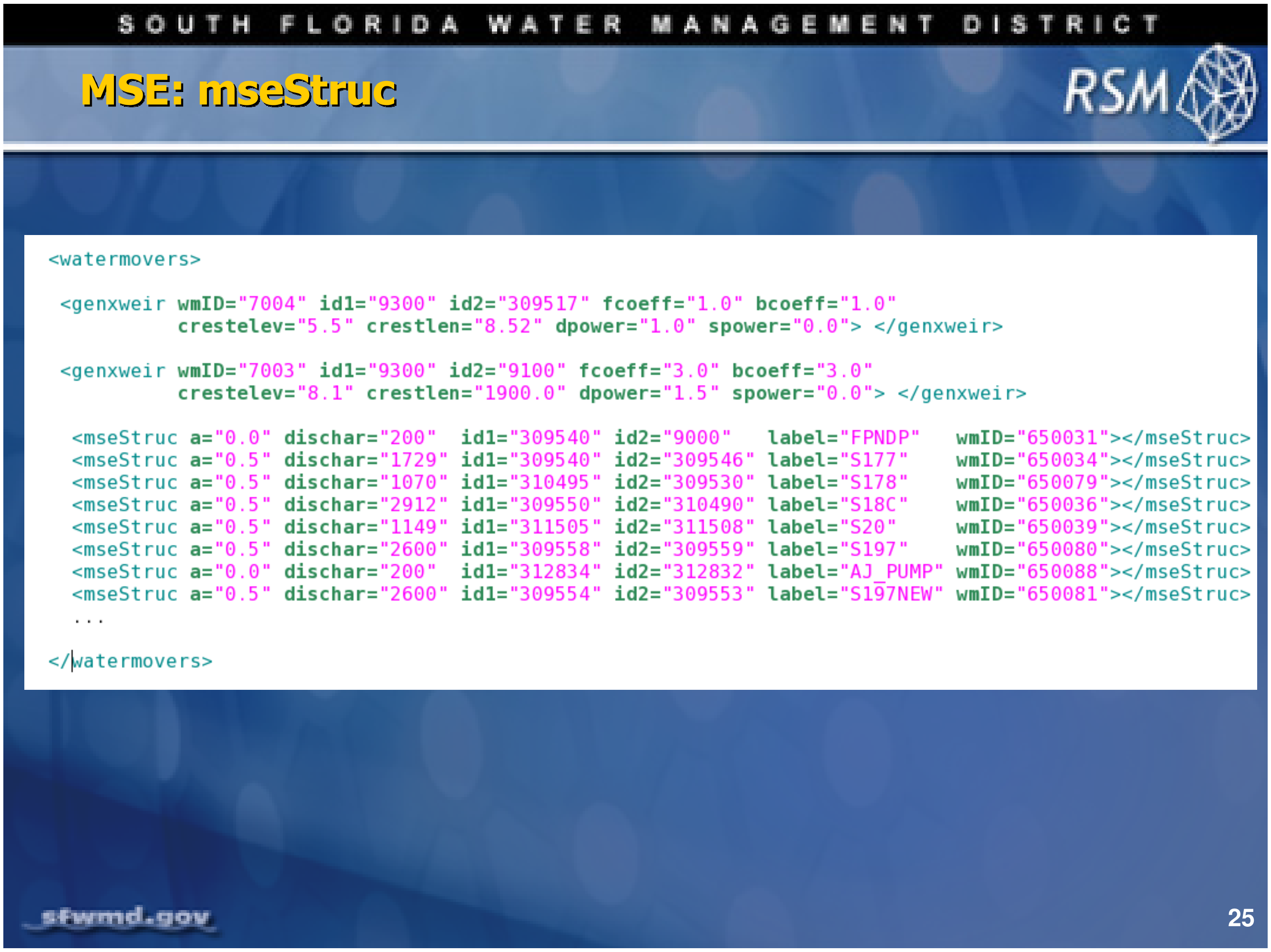
MSE: mse_nodes

RSM

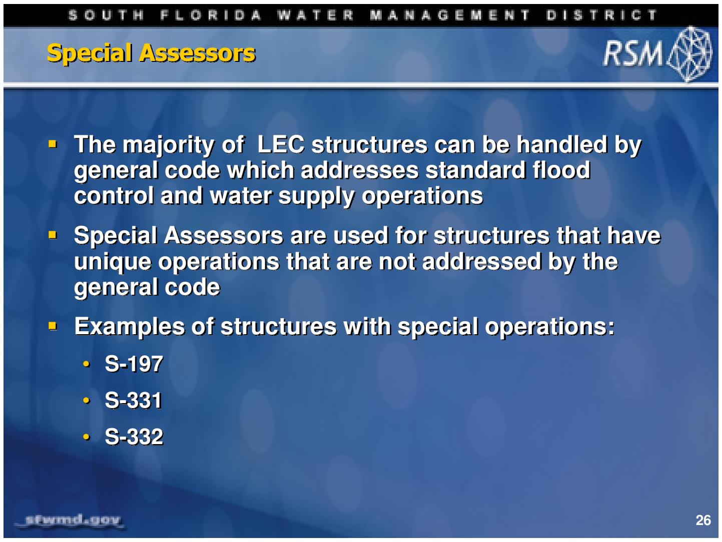
```
<mse_network name="MD-MSE Network" >
<version/>
   <mse_nodes>
      <mse_node designCap="2900" managed="yes" name="S177" purpose="wsfc" watermover="S177">
         <open  name="S177 Open">  <rc id="79503"></rc> </open>
         <close name="S177 Close"> <rc id="79504"></rc> </close>
      </mse_node>

      <mse_node designCap="200" managed="yes" name="FPNDP" purpose="fc" watermover="FPNDP">
         <open  name="FPNDP Open">  <rc id="79506"></rc> </open>
         <close name="FPNDP Close"> <rc id="79507"></rc> </close>
          <twHeadLimit name="FPNDP twHeadLimit"> <const value="7.5"> </const> </twHeadLimit>
          <offTrigger name="offtrigger" id="1004"> <rc id="79586"></rc>/> </offTrigger>
      </mse_node>

      <mse_node designCap="1275" managed="yes" name="S178" purpose="fc" watermover="S178">
         <open  name="S178 Open">  <rc id="791101"></rc> </open>
         <close name="S178 Close"> <rc id="791102"></rc> </close>
      </mse_node>

      <!-- S18C designcap originally 3200 cfs -->
      <mse_node designCap="2400" managed="yes" name="S18C" purpose="wsfcenv" watermover="S18C">
         <open  name="S18C Open">  <rc id="79603"></rc> </open>
         <close name="S18C Close"> <rc id="79604"></rc> </close>
         <minflow name="S18C minflow" purpose="env"> <rc id="79606"></rc>
         </minflow>
      </mse_node>
```

sfwmd.gov 24

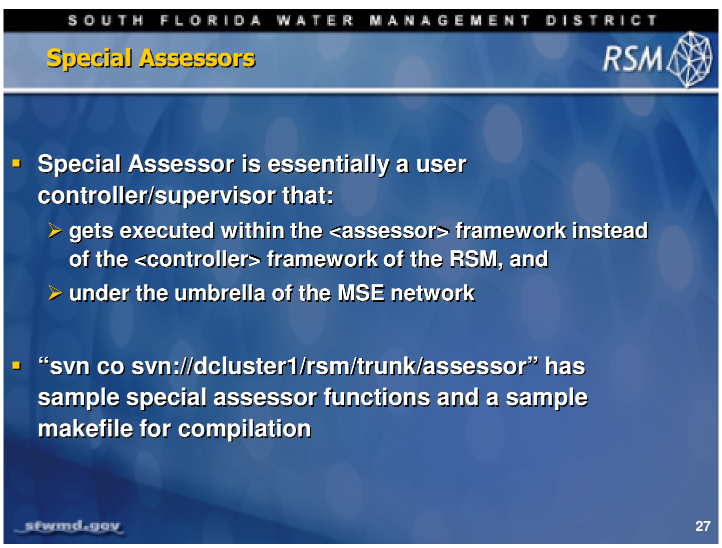
This is a typical implement of the mse_nodes. Each node identifies the associated structure, purpose and design capacity. In the minimum the open and closed levels are defined. For these three mse_nodes the open and closed values are defined by rule curves. The values could be set as constants or time series. In these cases the rule curves define constant values. Typically, the rule curves are used to define seasonal or monthly targets. The mse_node for the FPNDP (Flood Protection pump) watermover has additional tailwater level restriction and a season restriction on pump operation. The S18C mse_node sets a monthly minimum flow for environmental protection. Notice that the purpose for each structure is different, they all have flood control but S177 also has water supply.



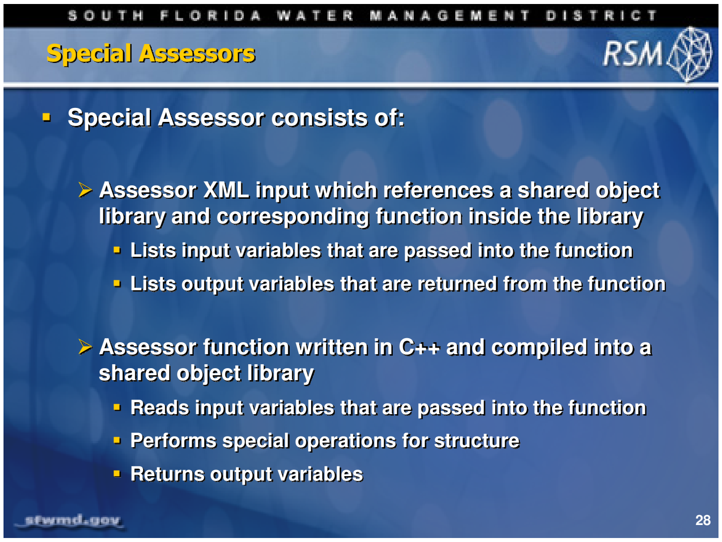
The flows determined by the MSE flood control, water supply and environmental assessors are implemented through the **<mseStruc>** watermovers. These watermovers define the discharge capacity of the structure and the mse_network determines the gain applied to the structure. As with all managed structures, these watermovers are controlled by controllers. The controllers are default controllers and visible to the user.



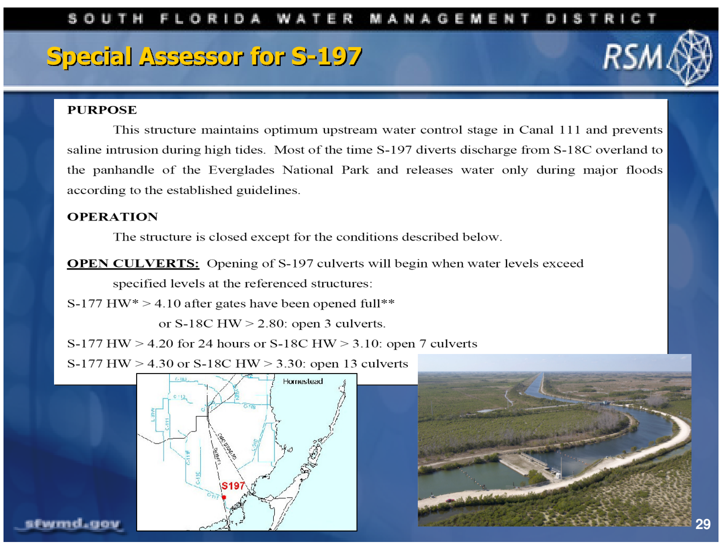
Occasionally there is a requirement for special assessors where the default assessors are not sufficient. Regional water management policies and practices can be implemented in the Lower East Coast Service Area (LECSA) using the standard RSM structures. This includes flood control that is managed using the upstream heads and water supply that is managed to maintain control levels in each WCU. The special assessors are necessary to control a few structures with complex water management. These include S-197, S-331 and S-332 in South Dade. It has not made sense to create generic controllers with these special requirements. There is no benchmark for the S-197 special assessor, but it can be seen in the C111 and other subregional models.



The special assessors are part of the MSE assessors so you need to have an MSE Network and you do not need standard controllers. It is necessary to check out the library of assessors. The source code is in that directory. There is a makefile for compiling the source code to create the shared library.

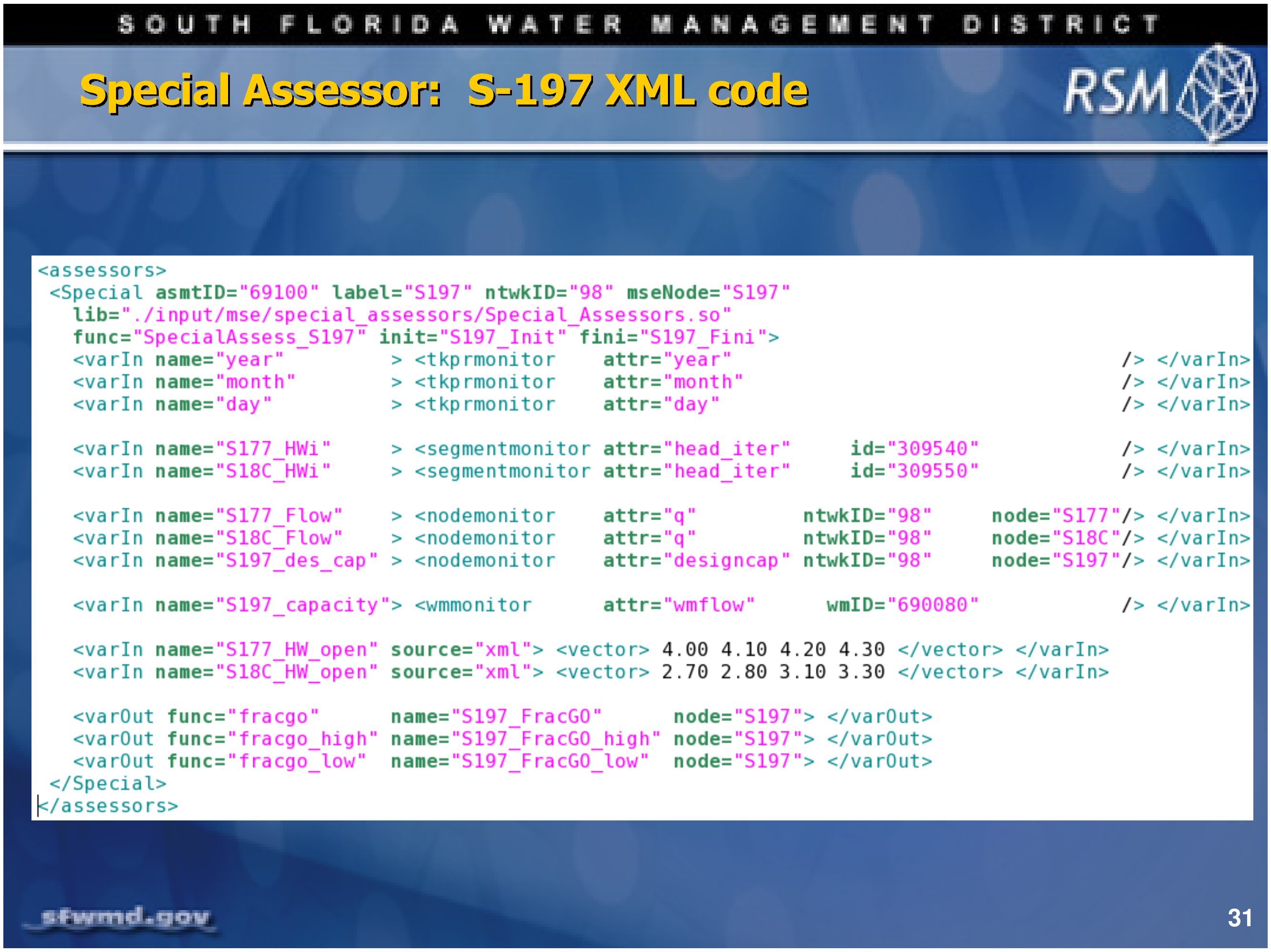


The Special Assessor has two components: the XML input and the compiled source code.



S-197 is a special structure located at the lower end of the canal system that is opened only to prevent serious flooding upstream. The structure has 13 culverts and the number of culverts that are opened depends on the head water levels at S-177 and S-18C structures located upstream. Since the structure operates more than one physical structure it was necessary to implement a special assessor to determine which culverts should be used.

The special assessor is defined in the run.xml file by the unique assessor identification number (asmtID), the network identification number (ntwkID), applicable MSE node (mseNode), the location of the shared function library (lib) and the function name in the shared library of compiled functions. The function may be written in Fortran, C or C++ language. The special assessor specification includes the list of input variables used by the special assessor and the output variables set by the assessor.




```xml
<assessors>
 <Special asmtID="69100" label="S197" ntwkID="98" mseNode="S197"
   lib="./input/mse/special_assessors/Special_Assessors.so"
   func="SpecialAssess_S197" init="S197_Init" fini="S197_Fini">
   <varIn name="year"         > <tkprmonitor    attr="year"                                    /> </varIn>
   <varIn name="month"        > <tkprmonitor    attr="month"                                   /> </varIn>
   <varIn name="day"          > <tkprmonitor    attr="day"                                     /> </varIn>

   <varIn name="S177_HWi"     > <segmentmonitor attr="head_iter"     id="309540"              /> </varIn>
   <varIn name="S18C_HWi"     > <segmentmonitor attr="head_iter"     id="309550"              /> </varIn>

   <varIn name="S177_Flow"    > <nodemonitor    attr="q"         ntwkID="98"     node="S177"/> </varIn>
   <varIn name="S18C_Flow"    > <nodemonitor    attr="q"         ntwkID="98"     node="S18C"/> </varIn>
   <varIn name="S197_des_cap" > <nodemonitor    attr="designcap" ntwkID="98"     node="S197"/> </varIn>

   <varIn name="S197_capacity"> <wmmonitor      attr="wmflow"      wmID="690080"             /> </varIn>

   <varIn name="S177_HW_open" source="xml"> <vector> 4.00 4.10 4.20 4.30 </vector> </varIn>
   <varIn name="S18C_HW_open" source="xml"> <vector> 2.70 2.80 3.10 3.30 </vector> </varIn>

   <varOut func="fracgo"      name="S197_FracGO"      node="S197"> </varOut>
   <varOut func="fracgo_high" name="S197_FracGO_high" node="S197"> </varOut>
   <varOut func="fracgo_low"  name="S197_FracGO_low"  node="S197"> </varOut>
 </Special>
</assessors>
```

This is an example of the application of the management control applied to an individual structure. This the management control of the S-197 structure. The management rules for the special assessor are written in C++ source code (presented below) and compiled in a shared library (SpecialAssessors.so). The various inputs for the assessor include the date (for seasonal stage requirements), head water stages, structure flows and rule curve vectors. The assessor can set the structure to zero flow (no operation), open 3 culverts, open seven culverts or open all 13 culverts.

SOUTH FLORIDA WATER MANAGEMENT DISTRICT

Special Assessor: S-197 function code

RSM

```
// NOTE: The SpecialAssess functions MUST return an int.
//       Return values are declared in mseIO.h as follows:
//          SPECIAL_FUNC_SUCCESS  1
//          SPECIAL_FUNC_FAILURE  0
using namespace std;
#include <iostream>
#include "../src/mseIO.h"
extern "C" int SpecialAssess_S197_FracGO( map<string, InputState*> *lpInputStateMap,
                map<string, OutputSpecialAssessor*> *lpOutputAssessorMap ) {
  string func = "SpecialAssess_S197_FracGO";
  try {
    double S177_HW_Stage = GetVarIn( func, "S177_HW_Stage", lpInputStateMap );
    double S177_Flow     = GetVarIn( func, "S177_Flow",     lpInputStateMap );

 // main code here..

       if ( not SpecialVarOut( func, "S197_FracGO",S197_FracGO,lpOutputAssessorMap ) {
          return SPECIAL_FUNC_FAILURE;
       }
       ...
       ...
       return SPECIAL_FUNC_SUCCESS;
  } // try

    catch(...) {
    cout << "ERROR: exception in function " << func << endl << flush ;
    return SPECIAL_FUNC_FAILURE ;
  }
}
```

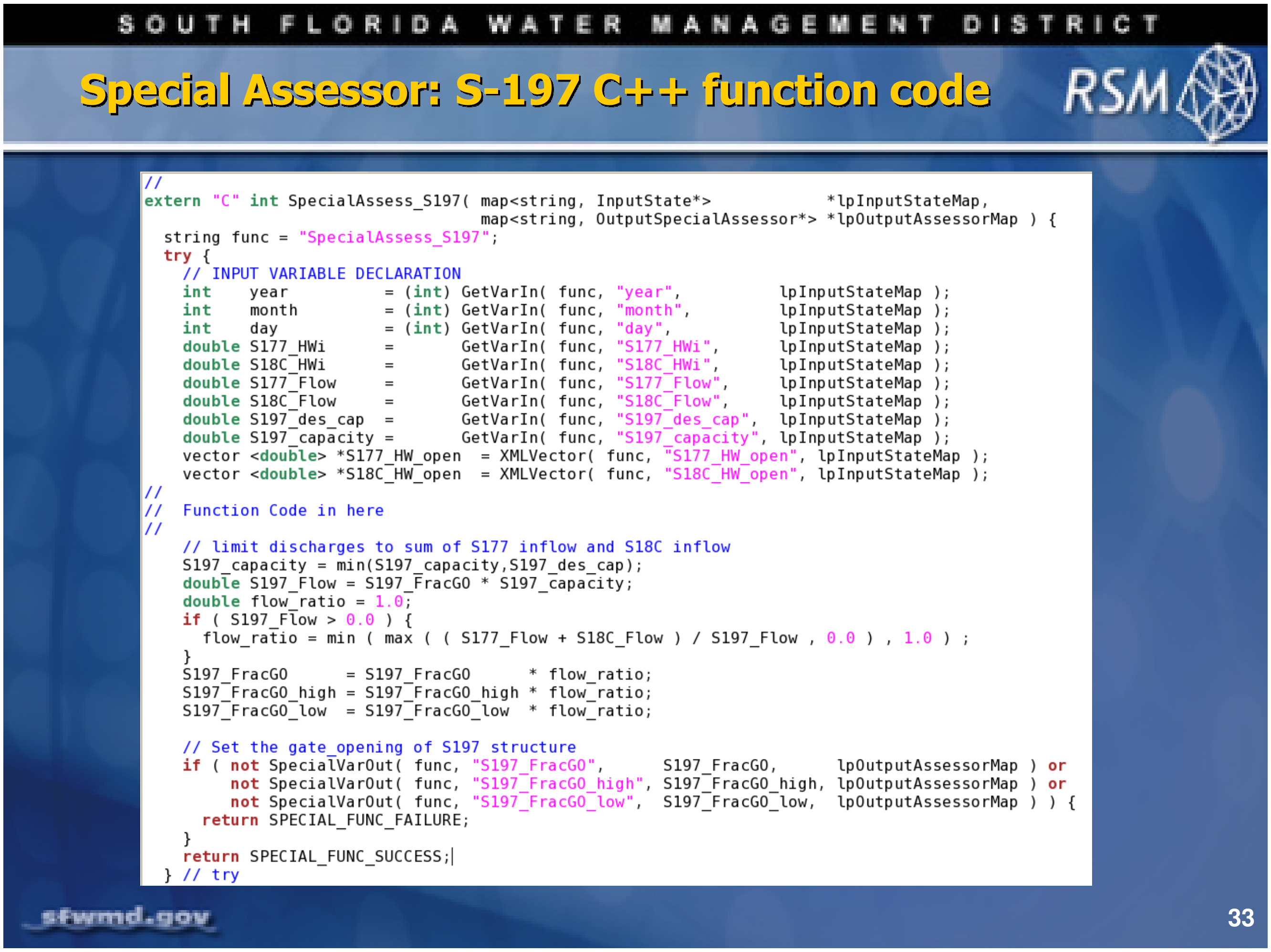
Inputs

Function code

Outputs

sfwmd.gov 32

This is the standard source code template for a special assessor. There are a few statements that are changed: 1) extern function name: “**SpecialAssess_S197_FracGO**”, 2) input variable names, 3) the main code that sets the output variables based on the input data, and 4) the return variable names. As with all function codes, there is error trapping to avoid unreasonable results.



The special assessor is defined by the first five attributes. The **varIn** provides the input data that the S-197 special assessor requires. The requirements for this special assessor come directly from the SFWMD Structure Book that describes the operations of the CSFFCP. S-197 has seven culverts which can be set to zero, three, seven or 13 culverts open. The headwater levels are provided in the vector above. Note: the return variables must be continuous. The return variables are a range of gate openings for each group of culverts.

SOUTH FLORIDA WATER MANAGEMENT DISTRICT

Special Assessor: C++ function code

RSM

```
// OUTPUT VARIABLE DECLARATION
double S197_FracGO = 0., S197_FracGO_high = 0., S197_FracGO_low = 0.;

if ( S177_HWi > S177_HW_open->at(3) || S18C_HWi > S18C_HW_open->at(3) ) {
  S197_FracGO = 13/13.;
}
else if ( S177_HWi > S177_HW_open->at(2) || S18C_HWi > S18C_HW_open->at(2) ) {
  S197_FracGO = 7/13. + (6/13.) *
      max( ( S177_HWi - S177_HW_open->at(2) ) / ( S177_HW_open->at(3) - S177_HW_open->at(2) ) ,
           ( S18C_HWi - S18C_HW_open->at(2) ) / ( S18C_HW_open->at(3) - S18C_HW_open->at(2) ) );
}
else if ( S177_HWi > S177_HW_open->at(1) || S18C_HWi > S18C_HW_open->at(1) ) {
  S197_FracGO = 3/13. + (4/13.) *
      max( ( S177_HWi - S177_HW_open->at(1) ) / ( S177_HW_open->at(2) - S177_HW_open->at(1) ) ,
           ( S18C_HWi - S18C_HW_open->at(1) ) / ( S18C_HW_open->at(2) - S18C_HW_open->at(1) ) );
}
else if ( S177_HWi > S177_HW_open->at(0) || S18C_HWi > S18C_HW_open->at(0) ) {
  S197_FracGO = (3/13.) *
      max( ( S177_HWi - S177_HW_open->at(0) ) / ( S177_HW_open->at(1) - S177_HW_open->at(0) ) ,
           ( S18C_HWi - S18C_HW_open->at(0) ) / ( S18C_HW_open->at(1) - S18C_HW_open->at(0) ) );
}
```

sfwmd.gov 34

This is a fragment of the C++ function code that is included in the special assessor code. It is a series of if-then-else statements that determine the number of gates opened as well as the flow ratio for each group of gates based on the water levels at S177 and S18C structures.

This function code can be altered and recompiled using the g++ compiler and added to the function library to create a new special assessor code for the structure.